Andmetüübid

Baasandmetüübid

Olulisemad andmetüübid:

```
- numeric (arvud)
- character (string/tekst)
- logical (tõeväärtused)
```

Lihtsaim viis mitme väärtuse koos hoidmiseks on vektor:

```
a <- c(1,2,5.3,6,-2,4) # numeric vector
b <- c("one","two","three") # character vector
c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) #logical vector
a
```

```
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
```

```
b
```

```
## [1] "one"   "two"   "three"
```

```
c
```

```
## [1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE
```

Arvud

Kõige lihtsamad asjad, mida Riga teha saab, on aritmeetilised operatsioonid.

```
1+2
```

```
## [1] 3
```

```
2*2
```

```
## [1] 4
```

```
1/2
```

```
## [1] 0.5
```

Muutujate väärtustamine

```
x<-5
y<-2
+ x
```

```
## [1] 5
```

```
- x
```

```
## [1] -5
```

```
x + y
```

```
## [1] 7
```

```
x - y
```

```
## [1] 3
```

```
x * y
```

```
## [1] 10
```

```
x / y
```

```
## [1] 2.5
```

```
#astendamine
x ^ y
```

```
## [1] 25
```

```
x ** y
```

```
## [1] 25
```

```
#modulo
x %% y
```

```
## [1] 1
```

```
#integer division
x %/% y
```

```
## [1] 2
```

Loogilised operatsioonid

```
arv1<-5
arv2<-11
arv1<arv2
```

```
## [1] TRUE
```

```
arv1<=arv2
```

```
## [1] TRUE
```

```
arv1==arv2
```

```
## [1] FALSE
```

```
boolean1=TRUE
boolean2=TRUE
boolean3=FALSE

! boolean1
```

```
## [1] FALSE
```

```
boolean1&boolean2
```

```
## [1] TRUE
```

```
boolean1&boolean3
```

```
## [1] FALSE
```

```
boolean1&&boolean2
```

```
## [1] TRUE
```

```
boolean1&&boolean3
```

```
## [1] FALSE
```

```
boolean1|boolean2
```

```
## [1] TRUE
```

```
boolean1|boolean3
```

```
## [1] TRUE
```

Komposiitandmetüübid

Vektor

```
#vektoriga oleme juba tuttavad, vektoriga saab teha tehteid
vec1=c(1,2,3,4)
vec2=c(10, 10,10,10)
vec3=c(20)
vec4=c(10,20)
vec5=c(10,20, 30)

vec1+vec2
```

```
## [1] 11 12 13 14
```

```
vec1+vec3
```

```
## [1] 21 22 23 24
```

```
vec1+vec4
```

```
## [1] 11 22 13 24
```

```
vec1+vec5
```

```
## Warning in vec1 + vec5: longer object length is not a multiple of shorter
## object length
```

```
## [1] 11 22 33 14
```

```
vec1=c(TRUE,FALSE, TRUE)
vec2=c(FALSE,FALSE, TRUE)
vec1+vec2
```

```
## [1] 1 0 2
```

```
vec1=c("tere", "halloo")
vec2=c("maja", "auto")
#mis võib olla tulemuseks?
#vec1+vec2
```

Elementidele viitamine:

```
vec=c(1,2, "auto", TRUE)
vec[1]
```

```
## [1] "1"
```

```
vec[1:length(vec)]
```

```
## [1] "1"    "2"    "auto" "TRUE"
```

```
vec[1:2]
```

```
## [1] "1" "2"
```

```
vec[c(1,2)]
```

```
## [1] "1" "2"
```

Andmeraam (dataframe)

Põhilisemaid andmetüüpe Ris.

```
d <- c(1,2,3,4)
e <- c("red", "white", "red", NA)
f <- c(TRUE,TRUE,TRUE,FALSE)
mydata <- data.frame(d,e,f)
names(mydata) <- c("ID","Color","Passed") # variable names
mydata
```

```
##   ID Color Passed
## 1  1   red   TRUE
## 2  2 white   TRUE
## 3  3   red   TRUE
## 4  4  <NA>  FALSE
```

Elementidele viitamine:

```
mydata[1:2] # veerud 3,4,5
```

```
##   ID Color
## 1  1   red
## 2  2 white
## 3  3   red
## 4  4  <NA>
```

```
mydata[c("ID","Color")] # veerud ID ja Color
```

```
##   ID Color
## 1  1   red
## 2  2 white
## 3  3   red
## 4  4  <NA>
```

```
mydata$Passed # muutuja Passed
```

```
## [1]  TRUE  TRUE  TRUE FALSE
```

```
mydata[1,2] #rida 1, veerg2
```

```
## [1] red
## Levels: red white
```

```
mydata[c(1,2), c(2,3)]
```

```
##   Color Passed
## 1   red   TRUE
## 2 white   TRUE
```

```
mydata[1:2, 2:3]
```

```
##   Color Passed
## 1   red   TRUE
## 2 white   TRUE
```

```
ncol(mydata)
```

```
## [1] 3
```

```
nrow(mydata)
```

```
## [1] 4
```

```
mydata
```

```
##   ID Color Passed
## 1  1   red   TRUE
## 2  2 white   TRUE
## 3  3   red   TRUE
## 4  4  <NA>  FALSE
```

```
mydata[2,3]<-"uus väärtus"
mydata
```

```
##   ID Color      Passed
## 1  1   red        TRUE
## 2  2 white uus väärtus
## 3  3   red        TRUE
## 4  4  <NA>       FALSE
```

Andmeraam kui vektorite kogum:

```
mydata['uus_veerg']<-c(25,10,20,30)
mydata
```

```
##   ID Color      Passed uus_veerg
## 1  1   red        TRUE        25
## 2  2 white uus väärtus        10
## 3  3   red        TRUE        20
## 4  4  <NA>       FALSE        30
```

```
mydata<-rbind(mydata,(rep("uus", ncol(mydata))))
```

```
## Warning in `[<-.factor`(`*tmp*`, ri, value = "uus"): invalid factor level,
## NA generated
```

```
mydata
```

```
##    ID Color      Passed uus_veerg
## 1   1   red        TRUE        25
## 2   2 white uus väärtus        10
## 3   3   red        TRUE        20
## 4   4  <NA>       FALSE        30
## 5 uus  <NA>         uus       uus
```

Funktsioonid rbind ja cbind lisavad vastavalt kas rea või veeru:

```
mydata <- mydata[-nrow(mydata),]
mydata
```

```
##   ID Color      Passed uus_veerg
## 1  1   red        TRUE        25
## 2  2 white uus väärtus        10
## 3  3   red        TRUE        20
## 4  4  <NA>       FALSE        30
```

```
mydata[-1,]
```

```
##   ID Color      Passed uus_veerg
## 2  2 white uus väärtus        10
## 3  3   red        TRUE        20
## 4  4  <NA>       FALSE        30
```

Loogilised operatioonid alamhulga leidmiseks:

```
mydata[ which(mydata$uus_veerg>10& mydata$Passed==TRUE), ]
```

```
##   ID Color Passed uus_veerg
## 1  1   red   TRUE        25
## 3  3   red   TRUE        20
```

```
# | - logical or
subset(mydata, uus_veerg>10| mydata$Passed==TRUE,select=c(Color, Passed))
```

```
##   Color Passed
## 1   red   TRUE
## 3   red   TRUE
## 4  <NA>  FALSE
```

Veel võimalusi alamhulkade leidmiseks.

```
subset(mydata, uus_veerg>10 & mydata$Passed==TRUE,select=c(Color, Passed))
```

```
##   Color Passed
## 1   red   TRUE
## 3   red   TRUE
```

```
newdata <- mydata[c(-3,-5)]
newdata
```

```
##   ID Color uus_veerg
## 1  1   red        25
## 2  2 white        10
## 3  3   red        20
## 4  4  <NA>        30
```

Maatriks

```
# generates 5 x 4 numeric matrix
y<-matrix(1:20, nrow=5,ncol=4)
y
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```
# another example
cells <- c(1,26,24,68)
rnames <- c("R1", "R2")
cnames <- c("C1", "C2")
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,
  dimnames=list(rnames, cnames))
mymatrix
```

```
##    C1 C2
## R1  1 26
## R2 24 68
```

```
y[,4] # 4th column of matrix
```

```
## [1] 16 17 18 19 20
```

```
y[3,] # 3rd row of matrix
```

```
## [1]  3  8 13 18
```

```
y[2:4,1:3] # rows 2,3,4 of columns 1,2,3
```

```
##      [,1] [,2] [,3]
## [1,]    2    7   12
## [2,]    3    8   13
## [3,]    4    9   14
```

Lists

An ordered collection of objects (components). A list allows you to gather a variety of (possibly unrelated) objects under one name.

```
# example of a list with 4 components -
# a string, a numeric vector, a matrix, and a scaler
w <- list(name="Fred", mynumbers=a, mymatrix=y, age=5.3)
w
```

```
## $name
## [1] "Fred"
##
## $mynumbers
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
##
## $mymatrix
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
##
## $age
## [1] 5.3
```

```
# example of a list containing two lists
v <- c(w,w)
v
```

```
## $name
## [1] "Fred"
##
## $mynumbers
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
##
## $mymatrix
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
##
## $age
## [1] 5.3
##
## $name
## [1] "Fred"
##
## $mynumbers
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
##
## $mymatrix
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
##
## $age
## [1] 5.3
```

```r
v$name
```

```
## [1] "Fred"
```

```r
v[["name"]]
```

```
## [1] "Fred"
```

```r
v[1]
```

```
## $name
## [1] "Fred"
```

```r
v[[1]]
```

```
## [1] "Fred"
```

```r
v$mymatrix[1:2,3:4]
```

```
##      [,1] [,2]
## [1,]   11   16
## [2,]   12   17
```